

REMARKS

Applicant and applicant's attorney express appreciation to the Examiner for the courtesies extended during the recent Examiner Interview held on January 17, 2003. The claim amendments made by this paper are consistent with the proposals discussed, and the agreements reached, during the interview.

The Office Action dated November 5, 2002 rejected Claim 15 under 35 U.S.C. 112, first paragraph indicating that "the best mode contemplated by the inventor has not been disclosed". Although Applicants disagree, Claim 15 has been cancelled herein thereby rendering this rejection moot.

The Office Action also rejected the remaining claims (Claims 1-14 and 16-43) under 35 U.S.C. 103(a) as being unpatentable over Ecklund (U.S. Pat. No. 4,853,843) in view of one or more other references.

Ecklund teaches a "system for merging virtual partitions of a distributed database" (see title). Each virtual partition "is a collection of sites that have access to a copy of the database and which can still communicate with each other" (Col. 2, lines 28-30). Such virtual partitions may be the result of some "system component failure" (Col. 2, line 26). "Each virtual partition independently executes group updates, each group update carrying out at least one or a set of operations with respect to a group of objects referenced by a configuration specification" (Col. 3, lines 16-19). "Each virtual partition maintains a separate change list describing all group updates that it executes" (Col. 3, lines 22-24).

Once the system failure that resulted in the virtual partition is corrected, the databases are merged. "To form the merged database, a virtual partition first obtains the change lists maintained by the separate virtual partitions. The partition then selects a collection of group updates from among all group updates described by the change lists" (Col. 3, lines 32-36) "subject to a restriction that the collection cannot include group updates described by different change lists of the separate virtual partitions when the group updates alter the same data object path" (Col. 3, lines 39-42). In other words, conflicting updates are not included in the collection. "The resolution algorithm manages group update conflicts by . . . resolving conflicting updates by **creating alternate versions**" (Col. 43, lines 65-68, emphasis added). Accordingly, Ecklund appears to resolve conflicts by storing both versions, a primary and an alternate version.

In contrast, Claim 1 (as amended herein) recites a method resolving a resource conflict in a system capable of replicating a server copy of a resource stored on one or more servers with a client copy of the resource stored on one or more clients. The method includes a server detecting that the resource on the server conflicts with the copy of the resource on a client, determining whether the server can resolve the conflict between the resource and the copy of the resource into a single version of the resource, and creating a conflict resource if the conflict cannot be resolved at the server. The client evaluates whether the conflict resource can be resolved into a single version of the resource in accordance with a schema of the client if the conflict was not resolved at the server. The user is then presented with the conflict resource conflict resource cannot be resolved by the client.

During the Examiner Interview, the Examiner agreed that Claim 1 appeared to be patentable over the cited art of reference, but that a further review of the rejection and the cited art would be required. In addition, the Examiner stated, an updated search would be required. Similar amendments have been made to the other remaining independent claims as were made to Claim 1. Ecklund does not teach the features of these other independent Claims for the same reasons that it does not teach the features of these other independent claims. Likewise, the other secondary references used to complete the 35 U.S.C. 103(a) do not teach or suggest all of the features of the independent claims that Ecklund does not teach. Accordingly, all of the independent claims are patentable over the combination of Ecklund and the other cited references. Accordingly, the pending claims are allowable and in patentable form. Favorable action is therefore requested.

In the event that the Examiner finds remaining impediment to a prompt allowance of this application that may be clarified through a telephone interview, the Examiner is requested to contact the undersigned attorney.

Dated this 21th day of January, 2003.

Respectfully submitted,



ADRIAN J. LEE

Attorney for Applicant

Registration No. 42,785



022913

PATENT TRADEMARK OFFICE

WA13768\119\DS0000000209V001.DOC

VERSION WITH MARKINGS TO SHOW CHANGES MADE
(09/412,738)

In the claims:

1. (Amended) In a system capable of replicating a server copy of a resource stored on one or more servers with a client copy of the resource stored on one or more clients, a method for resolving a resource conflict comprising the steps of:

detecting, by the server, that the resource on the server conflicts with the copy of the resource on a client;

determining, at the server, whether the server can resolve the conflict between the resource and the copy of the resource [can be resolved] into a single version of the resource;

creating, by the server, a conflict resource, if the conflict cannot be resolved at the server;

evaluating, at the client, whether the conflict resource can be resolved into a single version of the resource in accordance with a schema of the client if the conflict was not resolved at the server; and

presenting the conflict resource to a user if the conflict resource cannot be resolved by the client.

10. (Amended) In a system having multiple copies of a resource, a method for detecting and resolving a conflict between a client copy of the resource and a server copy of the resource, the method comprising the steps of:

receiving, from the client, a client resource tag at the server, wherein the client resource tag identifies a client version of the client copy of the resource;

determining, by the server, whether the client resource tag matches the server resource tag, wherein the server resource tag identifies a server version of the server copy of the resource;

determining that a conflict exists if the client resource tag does not match the server resource tag; and

VERSION WITH MARKINGS TO SHOW CHANGES MADE
(09/412,738)

executing a server level of conflict resolution between the client copy of the resource and the server copy of the resource at the server in order to resolve the server copy and client copy of the resource into a single version of the resource.

17. (Amended) In a system having one or more server copies of a resource and one or more client copies of a resource, a method for resolving a conflict between a server copy of the resource and a client copy of the resource into a single version of the resource, the method comprising the steps of:

receiving, from a server, a conflict resource at a client; and

executing a client level of conflict resolution between the client copy of the resource and the server copy of the resource at the client in order to resolve the server copy and client copy of the resource into a single version of the resource.

30. (Amended) In a system capable of replicating resources from one or more servers to one or more clients, a method for resolving a conflict, the method comprising the steps of:

detecting the conflict, wherein detecting the conflict comprises the steps of:

transmitting a client resource tag to a server;

comparing, by the server, the client resource tag with a server resource tag; and

determining that there is a conflict between a client copy of a resource and a server copy of the resource if the client resource tag does not match the server resource tag; and

executing one or more levels of conflict resolution until the conflict is resolved into a single version of the resource.

34. (Amended) A method as defined in claim 31, wherein the step of executing one or more levels further comprises the step of executing [a] the third level of conflict resolution.

VERSION WITH MARKINGS TO SHOW CHANGES MADE
(09/412,738)

35. (Amended) A method as defined in claim 30, wherein the step of executing one or more levels further comprises the step of resolving the conflict into the single version of the resource.

40. (Amended) In a system capable of replication a resource from one or more server to one or more clients, a computer program product for a method for detecting and resolving resource conflicts, the computer program product comprising:

a computer readable medium carrying computer executable instructions for implementing the method, wherein the computer executable instructions comprise:

program code means for detecting a resource conflict;

program code means for comparing a client resource tag with a server resource tag, wherein the client resource tag and the server resource tag are representative of a version of the resource;

program code means for resolving the conflict at the server into a single version of the resource;

program code means for creating a conflict resource if the conflict cannot be resolved at the server;

program code means for resolving the conflict at the client into a single version of the resource by evaluating the conflict resource in accordance with a schema known to the client; and

program code means for presenting the conflict resource to an end user for conflict resolution if the client cannot resolve the conflict .

42. (Amended) In a system capable or replicating a resource between a client and a server, a computer program product for a method for detecting and resolving a conflict between a client copy of the resource and a server copy of the resource, the computer program product comprising:

a computer readable medium carrying computer executable instructions for implementing the method, wherein the computer executable instructions comprise:

program code means for transmitting a client resource tag to a server;

VERSION WITH MARKINGS TO SHOW CHANGES MADE
(09/412,738)

program code means for comparing, by the server, the client resource tag with a server resource tag;

program code means for determining that there is a conflict between a client copy of a resource and a server copy of the resource if the client resource tag does not match the server resource tag; and

program code means for executing one or more levels of conflict resolution until the conflict is resolved into a single version of the resource.

43. (Amended) A computer program product as in claim 42, wherein the computer executable instructions further comprise program code means for:

executing a server level of conflict resolution to resolve a server copy of a resource and a client copy of a resource having one or more conflicts into a single version of the resource;

executing a client level of conflict resolution to resolve a server copy of a resource and a client copy of a resource having one or more conflicts into a single version of the resource; and

executing a third level of conflict resolution to thereby resolve a server copy of a resource and a client copy of a resource having one or more conflicts into a single version of the resource.